Bear Documentation

# Matlab Dependencies

* Deep Learning Toolbox
* Fuzzy Logic Toolbox
* Global Optimization Toolbox

# Known Bugs/Issues

* Nozzle size is assumed to match nozzle number (nozzle 0 = 0.75 mm nozzle, nozzle 1 = 0.5 mm Nozzle). Major changes to code would be needed to fix this problem.
  + Ex. Cylinder print type is selected by nozzle number, not nozzle size.
* Low density materials can only be loaded into large nozzles because there is not a cylinder gcode test for small nozzle low density.

# Main Variables

* Campaigns:
  + 1. Generalized Cylindrical Shells
  + 2. Beibits’s cylinder tests
  + 3. Squigly Prints (University of Washington Collaboration)
  + 4. Adedire’s Hexagons.
    - How to start campaign:
      1. Upload gcode to Campaign\_gcode folder
      2. Update ToPrintGcodeList
      3. Enable correct nozzle on printer
         1. Bear: enable nozzle: 4: 1
         2. Bear: disable nozzle: 4: 0
      4. Set stlmode to 13
         1. Bear: set stlmode: 4: 13
      5. Enable printer
         1. Bear: enable printer: 4: reset
    - How to edit ongoing campaign
      1. Disable printer
         1. Bear: disable printer: 4
      2. Download ToPrintGcodeList
      3. Make changes
      4. Upload ToPrintGcodeList back to drive
      5. Restart printer
         1. If part is printing or waiting to be picked up

Bear: enable printer: 4: resume

* + - * 1. Otherwise

Bear: enable printer: 4: reset

* printerT: Table of Cell Arrays that contains data relevant to the printers. A Cell can either be a 1 x 5 array or a 2 x 5 matrix. The column corresponds to the printer number. The row, if more than one, corresponds to the nozzle. Note that the nozzles are named 0 and 1, so when indexing in the matrix, it is often required to increment by 1 because matlab does not support index 0.
  + List of cells:
    - Status
      1. -2: printer disabled
         1. -2.1: Bed Not Clean
         2. -2.2: Consecutive Print Failures
         3. -2.3: Protective Stop
         4. -2.4: Manually Disabled
         5. -2.5: No active nozzles
         6. -2.6: Printer Connection Failure
      2. -1: (Deprecated)
      3. 0: Clean but waiting for data.
      4. 0.1: Clean and has data: ready to select next part
      5. 0.9: Part is being selected by secondary compute node
      6. 1: Waiting to generate STL
      7. 1.1: STL has been generated. Ready to slice and start the print.
      8. 2: printer is printing.
      9. 3: print has finished. Waiting to move part to scale.
      10. 4: no part: waiting to be cleaned, no data
      11. 4.1: no part: waiting to be cleaned, data

# Main Loop

## Generate STL

* selectExperiment Function
  + Must break out with return command. Otherwise, stays in while loop.
  + STL\_Mode
    - * 1. Cylinder test for modulus (when new filament is loaded in)
      * 2. Initial calibration test
      * 3. Looped calibration test
      * 6: ToPrintAndStore list. Prints a part, weighs it, then stores it.
      * 7: Repeated Print of Gcode.
      * 8. CLS prints
      * 11. Main campaign
      * 12. To print list
      * 13. To print from gcode list
      * 301. Clyinder Test for modulus
      * 302. Initial ext mult test
      * 303. Looped ext mult test
      * 311: Main campaign (campaign 3)
      * 312. To print from list (squiggly)
      * Note: Greater than 9 means main test campaign to include in GP
* decideExperiment Function
  + 1. Broad Steps:
       - Define samples that will make up GP (variable T)
       - Define x-inputs (xObsNorm)
       - Define y-inputs (yObs)
       - Define prediction x values (xPred)
       - Build GPR
       - Predict xPred using GPR
       - Assemble Prediction data into final form for evaluation
       - Apply Decision Function to GPR
    2. Decision Policy Options:
       - 1. Deprecated: Incorrect Thompson Sampling attempt
       - 2. Maximum Variance on Toughness/Mass (Originally just toughness)
       - 3. Expected Improvement on Toughness/Mass(Originally just toughness)
       - 4. Expected Improvement on Acceleration Model
       - 5. Expected Improvement as target stress (each part’s value at that stress level)
       - 6. Deprecated (Expected Improvement at target stress using critical points with gaussian penalty for distance from target stress)
       - 7. Deprecated (Same as 6 but added/corrected error propagation)
       - 8. Expected Improvement to move Composite Line Up
       - 9. Deprecated: EI on composite line with strict search range limits
       - 10. Expected Improvement to move Composite Line Up, but Composite Line is only parts printable by printer
       - 11. MV on EAE/Critical Stress for cylinders
       - 12: Expected Improvement on Max of Comp Line (with search range penalty)
       - 13: EI on EAE/Critical Stress for cylinders
       - 14: Pareto Front Random on max stress 20 vs. a model
       - 15: Pareto Front Random (max stress 20 vs. saved a model)
       - 16: Capped Parts EI on Crit Force \* KS model
       - 17: EI near Max part for loaded filament
       - 18: EI near best point
       - 19: UCB near best point
       - 20: UCB on max comp line with Restricted Relative Density
       - 21: Pareto Front Random on max stress 20 vs. Ft\*Ks
       - 22: EI on EAE/Critical Stress for Extrudable Parts
       - 23: EI on EAE/Critical Stress for Extrudable Parts with Linear Twist
       - 24: KSA: efficiency adjustment for 2 parallel parts (Adedire)
       - 25: KSA: efficiency adjustment for 2 parallel parts with zoomed in GP near best part(Adedire)
       - 301: MV on modulus (Squiggly Prints)
       - 302: MV on dz error (Squiggly Prints)
    3. Xmode (see getSpaceTISC)
       - 1. Standard GCS
       - 2. Cylinder/Cone Mode
       - 3. Cap Mode
       - 4. Extrudable Mode
       - 5. Extrudable Mode with twist
       - 301: General Squiggly
    4. Changing GP Input Variable (Steps)
       - Add variable to xObs
       - Add variable to xPred (likely requires edits to getSpaceTISC)
       - Check filterObs function to see if any variable needs changing
       - Delete Hyperparameters if any saved
    5. Case 8: holistic bear line
       - yMax: highest Energy Absorption by stress level
       - compLine: composite line (performance vs. critical stress)
       - yObsOrig: Performance (fraction of theoretical max)
       - yObs: hypertangent transformation of Performance
       - yObs2: Adjusted Stress Values (adjusted by modulus to a power).
       - xObs: input space of previous tests
       - xObsNorm: input space normalized between 0-1 of previous tests (for each input)
       - xPred: input space of possible experiments printer can run
       - xPredNorm: xPred normalized between 0-1 (for each input)
       - gprMdl: GP for yObs (performance)
       - printabilityTransformation: Penalty modifier to discount proposed tests that are likely not printable.
       - gprMdl2: GP for yObs2 (Adjusted Stress Values)
       - yMu2: Predicted critical stress values that correspond to xPredNorm using gprMdl2 (note that adjustment based on Mod is immediately removed)
       - ySdv2: Standard deviation that corresponds to yMu2
       - yMu: Predicted Performance (hypertangent transformation is removed immediately after prediction)
       - ySdv: Standard deviation that corresponds to yMu
       - yMu2Index: The index of the critical stress that corresponds to the critical stress of yMu2
       - zMu: the predicted amount that the composite line will improve
       - zSdv: the combined standard deviation through error propagation

## Start Print

* checkPrinterStatus
  + printing
    1. 2: printer is printing
    2. 3: printer is not printing and bed temperature is above 95C (ready to remove part)
    3. 4: printer is not printing, but bed temperature is below 95C
* Calibrating Mass:
  + Precalibration steps (contained in selectExperiment.m)
    1. Print initial calibration part and get mass (case 2)
    2. if last print is within 5% of target mass (3.3 g), then stop calibration. Else, print next calibration sample and alter STL\_Length according to linear adjustments.
    3. When calibration stopped, get Final Filament Length from the slicer and divide by target mass. This gets the initial filament length/mass ratio, which is then used to calculate projected mass of part going forward.
  + During Campaign
    1. In mergeState.m, calculates target filament length using I tuning to change initialfilamentlength/massratio slowly over time.
    2. in startPrint, changes extrusion multiplier to match target filament length to projected filament length, as reported from slicer. Stops when final length is within 1% of target length.

## Weigh Part

* + scaleStatus
    - -1: Move part on scale to storage instead of Instron.
    - 1: scale empty
    - 3: part successfully weighed.
    - -8: Impact Part. Human will remove part and then scaleStatus will be reset to 0.
    - -9: Protective Stop. Wait for human to free and then do enable printer: resume command
  + dataT.Printable is set to -0.9 if no mass is registered
  + printerT.Status set to 4 if part successfully removed.
  + printerT.FailCount: If part transferred successfully, set to 0. Else, increment by 1

## Crush Part

* instronStatus
  + 0: needs cleaning
  + 1: Ready for part
  + 2: Part Testing
  + 3: Test done: part needs removal

## Remove Part from Instron

## Get Instron File and Calculate Metric

* Steps for starting new Instron File
  + Use bear: pause for day to stop Bear. This ensures that no instron files are pending.
  + Start new instron file using the current method.
  + Save old folder (SampleData\_1\_Exports) at a batch#
  + Reset instron number in matlab using the command instronCountOffset = instronCount -1
    - Note, make sure this is saved to the test.mat file before restarting.

## Clean Instron

## Clean Print Bed

* Moves printerT.Status from 4 (needs cleaning) to 0 (ready for STL Generation)

## Monitor Print

## Main

* Adding Variable to dataT
  + Update dataT\_names
  + Update dataTadd size
  + Add new variable to dataT
  + Place new variable before Kernel Parameters

# Maintenance

## Windows updates

Windows will update when there is an update, even if you tell it not to. So, the best method is a two prong approach.

* Go to Windows update, click advanced options, then pause updates. Select “Pause until” and select the farthest date. Write this down in your calendar. Before this date, go back to windows updates and check for updates.
* If you see that windows has pushed an update (for example, your computer has an update), then update the BEAR computers immediately. Don’t trust the pause until if you know there is an update pending.